TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application No.	09/896,526
Filing Date	June 28, 2001
First Named Inventor	Haitham Akkary
Art Unit	2183
Examiner Name	David J. Huisman
Total Number of Pages in This Submission	30
Attorney Docket Number	42390P11201

ENCLOSURES *(check all that apply)*

- [X] Fee Transmittal Form
  - [X] Fee Attached
- [ ] Amendment / Response
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
  - [ ] PTO/SB/08
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Basic Filing Fee
  - [ ] Declaration/POA
- [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s)
  - [ ] Landscape Table on CD
- [ ] After Allowance Communication to TC
- [ ] Appeal Communication to Board of Appeals and Interferences
- [X] Appeal Communication to TC **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [X] Other Enclosure(s) *(please identify below):*

Return receipt postcard

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Steven Laut, Reg. No. 47,736 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	July 10, 2006

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	**Jean Svoboda**		
Signature		Date	**July 10, 2006**

Based on PTO/SB/21 (09-04) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 11/30/2005.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

FEE TRANSMITTAL for FY 2005

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/896,526
Filing Date	June 28, 2001
First Named Inventor	Haitham Akkary
Examiner Name	David J. Huisman
Art Unit	2183
Attorney Docket No.	42390P11201

METHOD OF PAYMENT *(check all that apply)*

☒ Check ☐ Credit card ☐ Money Order ☐ None ☐ Other (please identify): _____

☒ Deposit Account Deposit Account Number: 02-2666 Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☐ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee

☒ Charge any additional fee(s) or underpayment of fee(s) under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20. ☒ Credit any overpayments

FEE CALCULATION

1. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	31	- 35** =	0	X	50.00	=	$0.00
Independent Claims	4	- 4** =	0	X	200.00	=	$0.00
Multiple Dependent						=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	50	2202	25	Claims in excess of 20
1201	200	2201	100	Independent claims in excess of 3
1203	360	2203	180	Multiple Dependent claim, if not paid
1204	790	2204	395	**Reissue independent claims over original patent
1205	300	2205	150	**Reissue claims in excess of 20 and over original patent

***or number previously paid, if greater, For Reissues, see below*

SUBTOTAL (1) ($) 0.00

2. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
2053	130	2053	130	Non-English specification	
1251	120	2251	60	Extension for reply within first month	
1252	450	2252	225	Extension for reply within second month	
1253	1,020	2253	510	Extension for reply within third month	
1254	1,590	2254	795	Extension for reply within fourth month	
1255	2,160	2255	1,080	Extension for reply within fifth month	
1401	500	2401	250	Notice of Appeal	
1402	500	2402	250	Filing a brief in support of an appeal	500.00
1403	1,000	2403	500	Request for oral hearing	
1451	1,510	2451	1,510	Petition to institute a public use proceeding	
1460	130	2460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
1809	790	1809	395	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	790	2810	395	For each additional invention to be examined (37 CFR § 1.129(b))	
Other fee (specify)					

SUBTOTAL (2) ($) 500.00

SUBMITTED BY

Complete (if applicable)

Name (Print/Type)	Steven Laut	Registration No. (Attorney/Agent)	47,736	Telephone	(310) 207-3800
Signature				Date	07/10/06

Based on PTO/SB/17 (12-04) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 12/15/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Patent Application of:

Haitham Akkary and Sebastien Hily

Serial No. 09/896,526

Filed: June 28, 2001

For: **A MULTITHREADED PROCESSOR CAPABLE OF IMPLICIT MULTITHREADED EXECUTION OF A SINGLE-THREADED PROGRAM**

Examiner: David J. Huisman

Art Unit: 2183

APPEAL BRIEF

Mail Stop Appeal Brief-Patents
Commissioner for Patents
Post Office Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

Applicant submits, the following Amended Appeal Brief pursuant to 37 C.F.R. § 41.37 for consideration by the Board of Patent Appeals and Interferences. Applicant submits payment in the amount of $500.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 41.20(b)(2). This brief does not include any new or non-admitted amendments or any new or non-admitted affidavit or other evidence.

If necessary, the Commissioner is hereby authorized in this, concurrent and future replies, to charge payment or credit any overpayment to Deposit Account No. 02-2666 for any additional fees required under 37 C.F.R. §§ 1.16 or 1.17, particularly extension of time fees.

07/13/2006 MBELETE1 00000018 09896526
01 FC:1402 500.00 OP

I. REAL PARTY IN INTEREST

Haitham Akkary and Sebastien Hily, the parties named in the caption, assigned their rights to that disclosed in the subject application through an assignment recorded on June 28, 2001 (Reel/Frame 011963/0216) to Intel Corporation of Santa Clara, California. Thus, as owner at the time the brief is being filed, Intel Corporation of Santa Clara, California, is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-7, 9-11, 13, 15-22 and 24-35 are pending. Claims 8, 12, 14 and 23 are canceled. Claims 1-7, 9-11, 13, 15-22 and 24-35 are rejected. Claims 1-7, 9-11, 13, 15-22 and 24-35 are being appealed.

IV. STATUS OF AMENDMENTS

Applicant has not amended the claims subsequent to a final rejection.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Applicant submits below a concise explanation of the claimed subject matter defined in independent claims 1, 11, 20 and 29. Specific reference will be made to the lines in the paragraph (first line of the referenced paragraph is, therefore, considered line 1 regardless of the line number where the paragraph is located on the page).

Independent claim 1 describes

A. Independent Claim 1

Applicant's independent claim 1 relates to a device including a first processor 110 and a second processor 120 (see Fig. 1; page 2, paragraph [0012], lines 3-4). Processor 110 has a

scoreboard 214 and a decoder 211 (see fig. 2; page 4, paragraph [0015], line 2). Processor 120 has a scoreboard 324 and a decoder 321 (see Fig. 3, page, 4, paragraph [0015], lines 4-5). L1 cache 175, L2 cache 170, instruction cache 180 and data cache 190 are connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 6-8). A register buffer 130 is connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 4-5). A trace buffer 140 is connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 4-5). Memory instruction buffers (load buffer 150 and store buffer 160) are connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 5-6).

The first processor 120 and the second processor 110 perform single threaded applications using multithreading resources (see page 4 paragraph [0015], lines 8-11). That is, the two processors, working as a team, execute single threaded applications on a multi-thread environment) and do not have to convert the single thread to an explicit multi-thread. The first processor executes a single threaded application ahead of the second processor executing the same single threaded application to avoid misprediction (see page 5 paragraph [0017], lines 6-9). That is, first processor runs the thread and then directs the second processor to jump ahead at some program counter value by using the information already obtained by the first processor in the same single thread application (see page 4 paragraph [0016], lines 5-7).

The single threaded application contains the same number of instructions when executed on said first processor 110 and the second processor 120 (this is implicit; i.e., the same thread being executed by each processor). The single threaded application that is executed on the second processor avoids branch mispredictions from information received from the first processor (see page 5 paragraph [0017], lines 4-9). This is the result of the speculative processor (processor 120) jumping ahead in the program counter and giving the information to the commit processor (processor 110) before the commit processor reaches the same point in execution for which the speculative processor thread program counter points.

B. Independent Claim 11

Applicant's independent claim 11 relates to a process/method as illustrated in Fig. 7 (see also page 11 paragraph [0035], line 1-9, page 11 paragraph [0036], lines 1-10). The method

begins by executing instructions in a single thread by a first processor (Fig. 7, block 710). The instructions in the single thread are executed by a second processor as directed by the first processor (Fig. 7, block 740) see also page 5 paragraph [0016], lines 5-7). That is, the first processor instructs the second processor to jump ahead to a certain program counter in the single thread (page 5 paragraph [0016], lines 5-9). The second processor can execute instructions ahead of the first processor to avoid misprediction by the first processor (page 5 paragraph [0017], lines 6-9). The second processor tracks at least one register that is either loaded from a register file buffer or written by the second processor [Fig. 7, block 753; page 5 paragraph [0018], lines 9-11). The method then transmits control flow information from the second processor to the first processor (Fig. 7, block 754, page 5 paragraph [0017], lines 6-7). The first processor avoids branch prediction by receiving the control flow information (page 5 paragraph [0017], lines 7-9).

Next, the second processor transmits results to the first processor. The first processor avoids executing a portion of instructions by committing the results of the portion of instructions into a register file from a trace buffer (Fig. 7 block 760). The method clears a store validity bit and sets a mispredicted bit in a load entry in the trace buffer if a replayed store instruction has a matching store identification (ID) portion in a load buffer (block 760, page 7 paragraph [0024], lines 2-8).

The first processor and the second processor execute single threaded applications using multithreading resources (page 4 paragraph [0015], lines 7-10). The single thread is not converted to an explicit multiple-thread application (page 4 paragraph [0015], lines 7-11). The single threaded application contains the same number of instructions when executed on the first processor and the second processor (this is implicit; i.e., the same thread being executed by each processor).

C. Independent Claim 20

Applicant's independent claim 20 relates to a machine-readable medium containing instructions that are executed by a machine (e.g., a computer) (see page 12, paragraph [0038], lines 1-13). The instructions when executed, cause the machine to perform the following operations. Instructions in a single thread are executed by a first processor (Fig. 7, block 710). The instructions in the single thread are executed by a second processor as directed by the first

processor (Fig. 7, block 740) see also page 5 paragraph [0016], lines 5-7). That is, the first processor instructs the second processor to jump ahead to a certain program counter in the single thread (page 5 paragraph [0016], lines 5-9). The second processor can execute instructions ahead of the first processor to avoid misprediction by the first processor (page 5 paragraph [0017], lines 6-9). The second processor tracks at least one register that is either loaded from a register file buffer or written by the second processor [Fig. 7, block 753; page 5 paragraph [0018], lines 9-11). Control flow information is transmitted from the second processor to the first processor (Fig. 7, block 754, page 5 paragraph [0017], lines 6-7). The first processor avoids branch prediction by receiving the control flow information (page 5 paragraph [0017], lines 7-9).

Next, the second processor transmits results to the first processor. The first processor avoids executing a portion of instructions by committing the results of the portion of instructions into a register file from a trace buffer (Fig. 7 block 760). A store validity bit is cleared and a mispredicted bit in a load entry in the trace buffer is set if a replayed store instruction has a matching store identification (ID) portion in a load buffer (block 760, page 7 paragraph [0024], lines 2-8).The first processor and the second processor execute single threaded applications using multithreading resources (page 4 paragraph [0015], lines 7-10). The single thread is not converted to an explicit multiple-thread application (page 4 paragraph [0015], lines 7-11). The single threaded application contains the same number of instructions when executed on the first processor and the second processor (this is implicit; i.e., the same thread being executed by each processor).

D. Independent Claim 29

Applicant's independent claim 29 relates to a system including a first processor 110 and a second processor 120 (see Fig. 1; page 2, paragraph [0012], lines 3-4). Processor 110 has a scoreboard 214 and a decoder 211 (see fig. 2; page 4, paragraph [0015], line 2). Processor 120 has a scoreboard 324 and a decoder 321 (see Fig. 3, page, 4, paragraph [0015], lines 4-5). A bus 630 connected to the first processor 110 and the second processor 120. (Fig. 6, page 11 paragraph [0034], lines 1-3). A main memory 610 is also connected to the bus 630. (Fig. 6, page 11 paragraph [0034], lines 1-3).

L1 cache 175, L2 cache 170, instruction cache 180 and data cache 190 are connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 6-8). A register buffer 130 is connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 4-5). A trace buffer 140 is connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 4-5). Memory instruction buffers (load buffer 150 and store buffer 160) are connected to the first processor 110 and the second processor 120 (see Fig. 1, page 2 paragraph [0012], lines 5-6).

The first processor 120 and the second processor 110 perform single threaded applications using multithreading resources (see page 4 paragraph [0015], lines 8-11). That is, the two processors, working as a team, execute single threaded applications on a multi-thread environment) and do not have to convert the single thread to an explicit multi-thread. The first processor executes a single threaded application ahead of the second processor executing the same single threaded application to avoid misprediction (see page 5 paragraph [0017], lines 6-9). That is, first processor runs the thread and then directs the second processor to jump ahead at some program counter value by using the information already obtained by the first processor in the same single thread application (see page 4 paragraph [0016], lines 5-7).

The single threaded application contains the same number of instructions when executed on said first processor 110 and the second processor 120 (this is implicit; i.e., the same thread being executed by each processor). The single threaded application that is executed on the second processor avoids branch mispredictions from information received from the first processor (see page 5 paragraph [0017], lines 4-9). This is the result of the speculative processor

(processor 120) jumping ahead in the program counter and giving the information to the commit processor (processor 110) before the commit processor reaches the same point in execution for which the speculative processor thread program counter points.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. The Patent Office rejects claims 1-7, 9-10 and 29-35 under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy et al. ("Slipstream Processors: Improving both Performance and Fault Tolerance", ASPLOS, pp. 257-268, Nov. 2000) ("Sundaramoorthy") in view of U. S. Patent No. 6,757,811 issued to Mukherjee ("Mukherjee") in view of Hennessy and Patterson ("Computer Architecture A Quantitative Approach", Morgan Kaufmann, 1996) ("Hennessy").

B. The Patent Office rejects claims 11, 13, and 15-19 under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy and in further view of Akkary (WO 99/31594).

C. The Patent Office rejects claims 20-22, and 24-28 under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy in view of Akkary, as applied above, and in further view of Tanenbaum "Structured Computer Organization," Prentice-Hall, 1984, pp. 10-12 ("Tanenbaum").

Applicant presents these grounds of rejection for review.

VII. ARGUMENT

Applicant submits the following argument:

The following discussion sets forth in detail Applicant's analysis with respect to the patentability of claims 1-7, 9-10 and 29-35.

A. It is asserted in the Office Action that claims 1-7, 9-10 and 29-35 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy et al. ("Slipstream Processors: Improving both Performance and Fault Tolerance", ASPLOS, pp. 257-268, Nov. 2000) ("Sundaramoorthy") in view of U. S. Patent No. 6,757,811 issued to Mukherjee ("Mukherjee") in view of Hennessy and Patterson ("Computer Architecture A Quantitative Approach", Morgan Kaufmann, 1996) ("Hennessy"). Applicant respectfully traverses the aforementioned rejections for the following reasons.

According to MPEP §2142

> [t]o establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)).

Further, according to MPEP §2143.03, "[t]o establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. (In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)." "*All words in a claim must be considered* in judging the patentability of that claim against the prior art." (In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970), emphasis added.)

Applicant's claim 1 contains the limitations of

> [a]n apparatus comprising: a first processor and a second processor each having a scoreboard and a decoder; a plurality of memory devices coupled to the first processor and the second processor; a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer; a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and a plurality of memory instruction buffers coupled to the first processor and the second processor; wherein the first processor and the second processor perform single threaded applications using multithreading resources, and the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant's claim 29 contains the limitations of

> [a] system comprising: a first processor and a second processor each having a scoreboard and a decoder; a bus coupled to the first processor and the second processor; a main memory coupled to the bus; a plurality of local memory devices coupled to the first processor and the second processor; a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer; a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and a plurality of memory instruction buffers coupled to the first processor and the second processor, wherein the first processor and the second processor perform single threaded applications using multithreading resources, the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Sundaramoorthy discloses a multiprocessor system that executes two (i.e., multiple streams/threads) pseudo-redundant programs on separate processors on the same chip. The two programs have a differing amount of instructions. (Sundaramoorthy, column 2, lines 34-56). That is, one of the programs has more instructions than the other. (Sundaramoorthy, page 258, first column, lines 40-55). Sundaramoorthy further discloses that the A-stream has fewer instructions than the R-stream, which receives information from the A-stream. Both programs run in parallel on two processors. Sundaramoorthy, however, does not teach, disclose or suggest "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

In other words, Sundaramoorthy executes multiple-streams on two separate processors where each stream has a different number of instructions, as opposed to Applicant's claimed

invention, which executes a single thread application without converting or duplicating the single thread application to a multiple-thread application when using multiple-thread processing resources, where the single thread application is executed on two processors without changing the number of instructions (i.e., an exact duplicate).

Hennessy discloses using score boarding to aid in allowing instructions to execute out of order. Sundaramoorthy, however, is directed to finding instructions that do not effect final program output and removes these instructions from a second stream, for example redundant instructions. Therefore, the combination of the two prior art documents would not result in Applicant's claimed invention. Further, Hennessy does not teach, disclose or suggest "the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor."

Mukherjee is relied upon for disclosing a multi-threaded processor executes two single threads, where one thread executes slightly ahead of the other for fault tolerance. The two threads are duplicates of one another. Mukherjee deals with multiple threads in a multi-threading processor, not single threaded processes. Mukherjee further asserts that the leading and trailing thread are executed on the same processor. Mukherjee does not teach, disclose or suggest

> the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Moreover, it is asserted in the Office Action that it would benefit Sundaramoorthy to run two streams having the same amount of instructions with one running ahead of the other. This

completely opposes the disclosure of Sundaramoorthy as one of ordinary skill in the art would know that streams of the same amount of instructions only adds latency.

Therefore, even if Sundaramoorthy were combined with Mukherjee and Hennessy, the resulting invention would still not include all of Applicant's claimed limitations. And, therefore, there would be no motivation to combine Sundaramoorthy, Mukherjee with Hennessy.

Moreover, by viewing the disclosures of Sundaramoorthy, Mukherjee and Hennessy, one can not jump to the conclusion of obviousness without impermissible hindsight. According to MPEP 2142,

> [t]o reach a proper determination under 35 U.S.C. 103, the examiner must step backward in time and into the shoes worn by the hypothetical 'person of ordinary skill in the art' when the invention was unknown and just before it was made. In view of all factual information, the examiner must then make a determination whether the claimed invention 'as a whole' would have been obvious at that time to that person. Knowledge of applicant's disclosure must be put aside in reaching this determination, yet kept in mind in order to determine the 'differences,' conduct the search and evaluate the 'subject matter as a whole' of the invention. The tendency to resort to 'hindsight' based upon applicant's disclosure is often difficult to avoid due to the very nature of the examination process. However, impermissible hindsight must be avoided and the legal conclusion must be reached on the basis of the facts gleaned from the prior art.

Applicant submits that without first reviewing Applicant's disclosure, no thought, whatsoever, would have been made to the limitations of

> the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Neither Sundaramoorthy, Mukherjee, Hennessy, and therefore, nor the combination of the three, teach, disclose or suggest the limitations contained in Applicant's amended claims 1, 20 and 29, as listed above. Since neither Sundaramoorthy, Mukherjee, Hennessy, nor the combination of the three teach, disclose or suggest all the limitations of Applicant's amended claims 1 and 29, claims 1 and 29 are not obvious over Sundaramoorthy in view of Mukherjee and Hennessy since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly or indirectly depend from amended claims 1 and 29, namely claims 2-7 and 9-10, and 30-35, respectively, would also not be obvious over Sundaramoorthy in view of Mukherjee and Hennessy for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 1-7, 9-10 and 29-35 are respectfully requested.

The following discussion sets forth in detail Applicant's analysis with respect to the patentability of claims 11, 13, and 15-19.

B. It is asserted in the Office Action that claims 11, 13, and 15-19 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of Hennessy and in further view of Akkary (WO 99/31594). Applicant respectfully traverses the aforementioned rejection for the following reasons.

Applicant's claim 11 contains the limitations of

> [a] method comprising: executing a plurality of instructions in a single thread by a first processor; executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor, transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and clearing a store validity bit and setting a

mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant's claims 13 and 15-19 directly depend on amended claim 11. Applicant's claim 11 contains similar limitations as claims 1, 20 and 29 above. Namely,

> a single thread by a first processor; executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor, transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information; transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Neither Sundaramoorthy, Mukherjee, Hennessy, and therefore, nor the combination of the three teach, disclose or suggest the limitations contained in Applicant's amended claim 11.

Akkary discloses a system for ordering loads and stores in a multi-threaded processor using load and store buffers. Applicant is well aware of Akkary as Akkary is owned by Applicant's Assignee. Akkary does not teach, disclose or suggest

> the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Therefore, combining Akkary with Sundaramoorthy, Mukherjee and Hennessy would still not result in Applicant's claim 11 as the combination would still run single threaded applications without converting to explicit multi-threaded applications, where the two single threaded processes are executed on different processors and the two single-threaded processes have the same amount of instructions and use multithreading resources.

Neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, and therefore, nor the combination of the four, teach, disclose or suggest the limitations contained in Applicant's claim 11, as listed above. Since neither Sundaramoorthy, Hennessy, Mukherjee, Akkary, nor the combination of the four, teach, disclose or suggest all the limitations of Applicant's claim 11, Applicant's claim 11 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy, and further in view of Akkary since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly depend from claim 11, namely claims 13 and 15-19, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and further in view of Akkary for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 11, 13, and 15-19 are respectfully requested.

The following discussion sets forth in detail Applicant's analysis with respect to the patentability of claims 20-22, and 24-28.

C. It is asserted in the Office Action that claims 20-22, and 24-28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Sundaramoorthy in view of Mukherjee in view of

Hennessy in view of Akkary, as applied above, and in further view of Tanenbaum "Structured Computer Organization," Prentice-Hall, 1984, pp. 10-12 ("Tanenbaum"). Applicant respectfully traverses the aforementioned rejections for the following reasons.

Applicant's claim 20 contains the limitations of

> [a]n apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising: executing a single thread from a first processor; executing said single thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor to avoid misprediction; tracking at least one register that is one of loaded from a first buffer, and written by said second processor, said tracking executed by said second processor, the first buffer being a register file buffer, and clearing a store validity bit and setting a mispredicted bit in a load entry in a second buffer if a replayed store instruction has a matching store identification (ID) portion, the second buffer being a trace buffer, wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Applicant has similar limitations above in section I(B) regarding Sundaramoorthy in view of Mukherjee, Hennessy and Akkary.

Tanenbaum is relied on for asserting that "any instruction executed by hardware can also be executed in software."

> the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Therefore, combining Tanenbaum with Akkary, Sundaramoorthy, Mukherjee and Hennessy would still not result in Applicant's claim 20 as the combination would still run single threaded applications without converting to explicit multi-threaded applications, where the two single threaded processes are executed on different processors and the two single-threaded processes have the same amount of instructions and use multithreading resources.

Neither Sundaramoorthy, Mukherjee, Hennessy, Akkary, Tanenbaum, and therefore, nor the combination of the five, teach, disclose or suggest the limitations contained in Applicant's claim 20, as listed above. Since neither Sundaramoorthy, Mukherjee , Hennessy, Akkary, Tanenbaum, nor the combination of the five, teach, disclose or suggest all the limitations of Applicant's claim 20, Applicant's claim 20 is not obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary and further in view of Tanenbaum since a *prima facie* case of obviousness has not been met under MPEP §2142. Additionally, the claims that directly or indirectly depend from claim 20, namely claims 21-22 and 24-28, would also not be obvious over Sundaramoorthy in view of Mukherjee, Hennessy and Akkary, and further in view of Tanenbaum for the same reason.

Accordingly, withdrawal of the 35 U.S.C. § 103(a) rejections for Claims 20-22, and 24-28 is respectfully requested.

CONCLUSION

Based on the foregoing, Applicant requests that the Board overturn the rejection of all pending claims and hold that all of the claims of the present application are allowable.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR, & ZAFMAN LLP

Dated: July 10, 2006

By: _____
Steven Laut, Reg. No. 47,736

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING:

I hereby certify that this correspondence is being deposited as First Class Mail, with sufficient postage, with the United States Postal Service in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, Post Office Box 1450, Alexandria, Virginia 22313-1450 on July 10, 2006.

_____
Jean Svoboda

VIII. CLAIMS APPENDIX

The claims involved in this Appeal are as follows:

Claim 1. (Previously Presented) An apparatus comprising:

a first processor and a second processor each having a scoreboard and a decoder;

a plurality of memory devices coupled to the first processor and the second processor;

a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer;

a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and

a plurality of memory instruction buffers coupled to the first processor and the second processor;

wherein the first processor and the second processor perform single threaded applications using multithreading resources, and the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, said single threaded application is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and the single threaded application executed on the second processor avoids branch mispredictions from information received from said first processor.

Claim 2. (Previously Presented) The apparatus of claim 1, wherein the memory devices comprise a plurality of cache devices.

Claim 3. (Original) The apparatus of claim 1, wherein the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.

Claim 4. (Previously Presented) The apparatus of claim 3, wherein each of the plurality of L0 data cache devices store exact copies of store instruction data.

Claim 5. (Original) The apparatus of claim 1, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load- ordering buffer.

Claim 6. (Original) The apparatus of claim 5, the at least one store forwarding buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag

portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

Claim 7. (Original) The apparatus of claim 6, the at least one load ordering buffer comprising a structure having a plurality of entries, each of the plurality of entries having a tag portion, an entry validity portion, a load identification (ID) portion, and a load thread ID portion.

Claim 8 (Canceled)

Claim 9. (Previously Presented) The apparatus of claim 1, wherein the trace buffer is a circular buffer.

Claim 10. (Original) The apparatus of claim 1, the register buffer comprising an integer register buffer and a predicate register buffer.

Claim 11. (Previously Presented) A method comprising:

executing a plurality of instructions in a single thread by a first processor;

executing said plurality of instructions in the single thread by a second processor as directed by the first processor, the second processor executing said plurality of instructions ahead of the first processor to avoid misprediction;

tracking at least one register that is one of loaded from a register file buffer, and written by said second processor, said tracking executed by said second processor,

transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information;

transmitting results from the second processor to the first processor, the first processor avoiding executing a portion of instructions by committing the results of the portion of instructions into a register file from a first buffer, the first buffer being a trace buffer, and

clearing a store validity bit and setting a mispredicted bit in a load entry in the first buffer if a replayed store instruction has a matching store identification (ID) portion in a second buffer, the second buffer being a load buffer,

wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and the single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Claim 12. (Cancelled)

Claim 13. (Previously Presented) The method of claim 11, further including:

duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

Claim 14. (Cancelled)

Claim 15. (Previously Presented) The method of claim 11, further including:

setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion in a load buffer.

Claim 16. (Previously Presented) The method of claim 11, further including:

flushing a pipeline, setting a mispredicted bit in a load entry in the trace buffer and restarting a load instruction if one of the load is not replayed and does not match a tag portion in a load buffer, and the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 17. (Previously Presented) The method of claim 11, further including:

executing a replay mode at a first instruction of a speculative thread.

Claim 18. (Previously Presented) The method of claim 11, further including:

supplying names from the trace buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOPs) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the trace buffer to the register file.

Claim 19. (Previously Presented) The method of claim 11, further including:

clearing a valid bit in an entry in a load buffer if the load entry is retired.

Claim 20. (Previously Presented) An apparatus comprising a machine-readable medium containing instructions which, when executed by a machine, cause the machine to perform operations comprising:

executing a single thread from a first processor;

executing said single thread from a second processor as directed by the first processor, the second processor executing instructions ahead of the first processor to avoid misprediction;

tracking at least one register that is one of loaded from a first buffer, and written by said second processor, said tracking executed by said second processor, the first buffer being a register file buffer, and

clearing a store validity bit and setting a mispredicted bit in a load entry in a second buffer if a replayed store instruction has a matching store identification (ID) portion, the second buffer being a trace buffer,

wherein the first processor and the second processor execute single threaded applications using multithreading resources, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Claim 21. (Original) The apparatus of claim 20, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

transmitting control flow information from the second processor to the first processor, the first processor avoiding branch prediction by receiving the control flow information.

Claim 22. (Original) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

duplicating memory information in separate memory devices for independent access by the first processor and the second processor.

Claim 23. (Cancelled)

Claim 24. (Original) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

setting a store validity bit if a store instruction that is not replayed matches a store identification (ID) portion.

Claim 25. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

flushing a pipeline, setting a mispredicted bit in a load entry in the second buffer and restarting a load instruction if one of the load is not replayed and does not match a tag portion in

a load buffer, and the load instruction matches the tag portion in the load buffer while a store valid bit is not set.

Claim 26. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

executing a replay mode at a first instruction of a speculative thread;

terminating the replay mode and the execution of the speculative thread if a partition in the second buffer is approaching an empty state.

Claim 27. (Previously Presented) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

supplying names from the second buffer to preclude register renaming;

issuing all instructions up to a next replayed instruction including dependent instructions;

issuing instructions that are not replayed as no-operation (NOPs) instructions;

issuing all load instructions and store instructions to memory;

committing non-replayed instructions from the second buffer to a register file.

Claim 28. (Original) The apparatus of claim 21, further containing instructions which, when executed by a machine, cause the machine to perform operations including:

clearing a valid bit in an entry in a load buffer if the load entry is retired.

Claim 29. (Previously Presented) A system comprising:

a first processor and a second processor each having a scoreboard and a decoder;

a bus coupled to the first processor and the second processor;

a main memory coupled to the bus;

a plurality of local memory devices coupled to the first processor and the second processor;

a first buffer coupled to the first processor and the second processor, the first buffer being a register buffer;

a second buffer coupled to the first processor and the second processor, the second buffer being a trace buffer; and

a plurality of memory instruction buffers coupled to the first processor and the second processor,

wherein the first processor and the second processor perform single threaded applications using multithreading resources, the first processor executes a single threaded application ahead of the second processor executing said single threaded application to avoid misprediction, and said single thread is not converted to an explicit multiple-thread application, said single threaded application contains the same number of instructions when executed on said first processor and said second processor, and said single threaded application executed on the second processor avoids branch mispredictions using information received from said first processor.

Claim 30. (Original) The system of claim 29, the local memory devices comprise a plurality of cache devices.

Claim 31. (Original) The system of claim 30, the first processor is coupled to at least one of a plurality of zero level (L0) data cache devices and at least one of a plurality of L0 instruction cache devices, and the second processor is coupled to at least one of the plurality of L0 data cache devices and at least one of the plurality of L0 instruction cache devices.

Claim 32. (Previously Presented) The system of claim 31, wherein each of the plurality of L0 data cache devices store exact copies of store instruction data.

Claim 33. (Original) The system of claim 31, the first processor and the second processor each sharing a first level (L1) cache device and a second level (L2) cache device.

Claim 34. (Original) The system of claim 29, wherein the plurality of memory instruction buffers includes at least one store forwarding buffer and at least one load ordering buffer.

Claim 35. (Original) The system of claim 34, the at least one store forwarding buffer including a structure having a plurality of entries, each of the plurality of entries having a tag portion, a validity portion, a data portion, a store instruction identification (ID) portion, and a thread ID portion.

IX. EVIDENCE APPENDIX

Applicant does not submit further evidence as the evidence relied on for the grounds of rejection pertain only to the cited prior art.

X. RELATED PROCEEDINGS APPENDIX

Applicant asserts there are no related proceedings that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.